What is claimed is:

1. A method for converting a filesystem from a first type to a second type, the method comprising the steps of:

generating a list of directories of the first type in the filesystem to convert; and

converting each directory in the list to the second type while maintaining the file system in an active state.

2. The method of claim 1, further comprising the step of:

sequentially initiating the steps of generating and converting upon initial program load of a computer system utilizing the filesystem.

3. The method of claim 1, wherein the step of converting further includes the steps of:

retrieving an identifier of a directory in the list;

converting the directory to a second-type directory; and

activating the second-type directory.

4. The method of claim 3, further comprising the step of:

repeating the steps of claim 3 for each directory in the list.

5. The method according to claim 1 wherein the list represents a top-down view of the filesystem spanning from a root directory down to an outermost leaf-node.

6. The method according to claim 5, wherein the step of converting each directory is performed for each directory in an order opposite to that of the list.

7. The method according to claim 1, wherein the step of generating further includes the steps of:

a) adding a root directory as a current entry in the list;

b) identifying a child directory of the current entry in the list;

c) appending the identified child directory to the list;

d) repeating steps b) and c) for each child directory within the current entry;

e) changing a next directory in the list immediately following the current entry to be the current entry, if the next directory exists in the list; and

f) repeating steps b) - e) until no next directory exists in the list.

8. The method of claim 1, further comprising the step of:

marking a particular directory as being in the process of conversion once the particular directory is in the list.

9. The method of claim 8, wherein a new object added to the particular directory is appended at an end of the particular directory.

10. The method of claim 1, wherein the step of converting further includes the steps of:

a) creating a second-type root directory;

b) creating a second-type directory corresponding to a particular directory in the list;

c) generating a respective link in the second-type directory for each child object of the particular directory;

d) activating the second-type directory; and

e) removing the particular directory from the list.

f) deleting the first-type directory

11. The method of claim 10, further comprising the step of:

creating a data structure associated with the second-type directory, the data structure including a first anchor point that is associated with a parent directory of the directory and a second anchor point associated with a parent directory of the second-type directory.

12. The method of claim 1, further comprising the steps of:

detecting a condition for pausing the converting step; and

in response to the condition, pausing the converting step.

13. The method of claim 12, wherein the condition is one of:

a product install on the filesystem; a restore operation involving the filesystem; and a back-up operation involving the filesystem.

14. The method of claim 1, further comprising the steps of:

determining a usage rate of a particular directory before converting that directory; and

postponing converting the particular directory based on the usage rate.

15. The method of claim 1, further comprising the steps of:

detecting a condition for stopping the converting step; and

in response to the condition, stopping the converting step.

16. The method of claim 15, wherein the condition is one of:

a system crash, encountering a corrupted object within the filesystem, and insufficient available storage.

17. The method of claim 15, wherein the converting step is restarted upon a subsequent initial program load involving the filesystem.

18. The method of claim 3, wherein the step of activating further includes the steps of:

identifying a data structure associated with the directory;

changing the data structure to be associated with the second-type directory; and

removing the directory.

19. The method of claim 18, further comprising the step of:

asserting a lock on first data structure while performing the step of changing.

20. A method for converting a filesystem from a first type to a second type, the method comprising the steps of:

executing a conversion process to convert each directory of the first type in the filesystem into the second type while maintaining the filesystem in an active state; and

terminating the conversion process when every directory of the first type in the filesystem has been converted to the second type.

21. The method of claim 20, further comprising the step of:

initiating the executing step upon initial program load of a computer system utilizing the filesystem.

22. The method of claim 20, further comprising the step of:

generating a list of directories of the first type in the filesystem to convert.

23. The method of claim 22, wherein converting each directory includes the steps of:

retrieving an identifier of a directory in the list;

converting the directory to a second-type directory; and

activating the second-type directory.

24. A program product, comprising:

a program code configured upon execution to:

generate a list of directories of a first type in a filesystem to convert; and

convert each directory in the list to a second type while maintaining the filesystem in an active state; and

a signal bearing medium bearing the program code.

25. A program product, comprising:

a program code configured upon execution to:

initiate a conversion process to convert each directory of a first type in a filesystem into a second type while maintaining the filesystem in an active state; and

terminate the conversion process when every directory of the first type in the filesystem has been converted to the second type; and

a signal bearing medium bearing the program code.

26. An apparatus comprising:

at least one processor;

a memory coupled with the at least one processor; and

a program code residing in memory and executed by the at least one processor, the program code configured to:

initiate a conversion process to convert each directory of a first type in a filesystem into a second type while maintaining the filesystem in an active state; and

terminate the conversion process when every directory of the first type in the filesystem has been converted to the second type.

27. The apparatus of claim 26, wherein the program code is further configured to:

non-recursively build a list of directories of the first type.

28. The apparatus of claim 26, wherein the program code is further configured to:

modify a particular directory being converted such that any new entries in that particular directory are appended at an end of that particular directory.

29. The apparatus of claim 26, wherein the program code is further configured to:

pause the conversion process if a predetermined operation is detected as being performed on the filesystem.

30. The apparatus of claim 26, wherein the program code is further configured to:

for a particular directory being converted, maintain a first version having the first type and a second version having the second type; and

activate the second version while de-activating the first version.

31. The apparatus of claim 26, wherein the program code is further configured to:

stop the conversion process at a particular point when an abnormal system condition is detected; and

restart the conversion process at the particular point.

32. The apparatus of claim 26, wherein the program code is further configured to:

for a particular directory being converted, determine if the particular directory is being used more than a predetermined amount; and

postpone converting that particular directory while it is being used more than the predetermined amount.

33. The apparatus of claim 32, wherein the program code is further configured to:

convert another directory, different than the particular directory, while the particular directory is being used more than the predetermined amount.

34. The apparatus of claim 26, wherein the program code is further configured to:

for a particular directory already converted, convert the particular directory back to the first type.

35. The apparatus of claim 26, wherein the program code is further configured to:

execute at an adjustable priority level.